

(51) Int.Cl. ⁹	識別記号	F I	
H 0 4 N 7/24		H 0 4 N 7/13	Z
G 0 6 T 1/60		5/455	
H 0 4 N 5/455		5/60	1 0 1
5/60	1 0 1	G 0 6 F 15/64	4 5 0 F
7/025		H 0 4 N 7/08	

審査請求 未請求 請求項の数12 O L 外国語出願 (全 34 頁) 最終頁に続く

(21) 出願番号 特願平9-277813

(22) 出願日 平成9年(1997)10月9日

(31) 優先権主張番号 8 6 1 0 9 1 0 2

(32) 優先日 1997年6月28日

(33) 優先権主張国 台湾 (TW)

(71) 出願人 595159677

聯華電子股▲ふん▼有限公司

台湾新竹科学工業園區工業東三路3號

(72) 発明者 ▲吳▼ 文義

台湾新竹縣竹北市光明十街58巷5號

(72) 発明者 潘 志新

台湾新竹縣竹東鎮北興路三段493-1號6樓

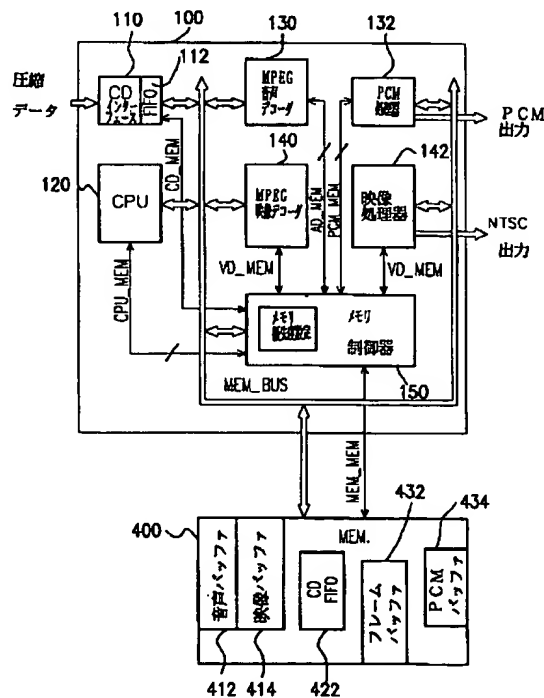
(74) 代理人 弁理士 秋元 輝雄

(54) 【発明の名称】 MPEGプロセッサにおけるメモリアクセス優先制御の適応性選択方法

(57) 【要約】

【課題】 メモリバスのバンド幅のバランスのとれた使用を達成するために、MPEG処理器内のメモリアクセスの優先度制御を行う適応性選択法を提供する。

【解決手段】 MPEG処理器100におけるメモリアクセス優先度制御の適応性選択法である。前記処理器は複数の機能モジュール、即ち入力インターフェース110、CPU120、音声デコーダ130、映像デコーダ140、音声処理器132、映像処理器142、およびメモリ制御器150を備える。前記複数のモジュールのそれぞれは、前記メモリ制御器による仲裁によってデータバスMEM-BUSに対するメモリアクセス用の制御を受ける。前記データバスに対する前記CPUのアクセス優先度は、前記CPUがMPEG圧縮データの解剖を実行する必要があるとき、および音声圧縮データの初期復号を実行するとき以外は、相対的に低く維持される。それ故、データバスのバンド幅の使用は、システム資源400の間でバランスされ、これによりシステム全体の性能が増大する。



【特許請求の範囲】

【請求項1】 MPEG処理器におけるメモリアクセス優先度制御の適合選択法であって、前記処理器は、圧縮データを受信し、そしてMPEG圧縮データを生成する入力インターフェースと、

前記MPEG圧縮データから音声及び映像圧縮データを解剖する中央処理ユニットと、

前記音声および映像圧縮データからそれぞれ音声および映像データを復号する音声デコードおよび映像デコードと、

前記音声および映像データからそれぞれ音声および映像解凍出力信号を生成する音声処理器および映像処理器と、およびデータバス上のアクセスを仲裁して、前記MPEGデータ、音声および映像圧縮データ、音声および映像データをメモリにストアするメモリ制御器とを備え、前記入力インターフェース、前記中央処理ユニット、前記音声および映像デコード、前記音声および映像処理器、および前記メモリ制御器は、相互間で通信できるように前記バス上で相互に接続されており、前記適合選択法は、無限ルーチンであり、そして前記中央処理ユニットが初期音声復号を実行する必要があるときは、前記データバスに対する前記中央処理ユニットのアクセス優先度を上昇させ、そして前記音声データの

前記初期復号の後は前記上昇したアクセス優先度を低下させるステップと、および前記中央処理ユニットが前記音声および映像圧縮データを解剖する必要があるときは、前記データバスに対する前記中央処理ユニットのアクセス優先度を上昇させ、また前記中央処理ユニットによる前記音声および映像圧縮データの解剖を実行させ、更に前記解剖した後は前記上昇したアクセス優先度を低下させるステップとを備えることを特徴とするメモリアクセス優先度制御の適合選択法。

【請求項2】 前記中央処理ユニットがMPEG圧縮データを解剖する必要があるときは、前記データバスに対する前記中央処理ユニットのアクセス優先度を上昇させ、また前記初期音声復号を実行させるステップは、更に、

前記音声圧縮データに対応する前記データおよび前記映像圧縮データに対応する前記データを生成する解剖を行うために、前記中央処理ユニットが前記メモリにストアされている前記MPEG圧縮データを前記データバスを介して検索するステップと、

前記音声圧縮データに対応する前記データを前記データバスを介して前記メモリにストアするステップと、

前記映像圧縮データに対応する前記データを初期復号して前記映像圧縮データを得るステップと備えることを特徴とする請求項1の適合選択法。

【請求項3】 前記中央処理ユニットがMPEG圧縮データを解剖する必要があるときは、前記データバスに対する前記中央処理ユニットのアクセス優先度を上昇さ

せ、また前記初期音声復号を実行させるステップは、更に、

前記音声圧縮データを生成する初期復号を行うために、前記中央処理ユニットが前記メモリにストアされている前記音声圧縮データに対応する前記データを前記データバスを介して検索するステップと、

前記データバスを介して前記音声圧縮データを前記メモリにストアするステップと備えることを特徴とする請求項2の適合選択法。

10 【請求項4】 前記入力インターフェースは、ビデオコンパクトディスクによって生成されたMPEG圧縮データを受信するためのCDインターフェースであることを特徴とする請求項3の適合選択法。

【請求項5】 前記入力インターフェースは、デジタルビデオディスクによって生成されたMPEG圧縮データを受信するためのCDインターフェースであることを特徴とする請求項3の適合選択法。

20 【請求項6】 前記CDインターフェースは、ビデオコンパクトディスクによって生成されたシリアル圧縮データをパラレルデータに変換するためのシリアル-パラレル変換器を更に備えることを特徴とする請求項4の適合選択法。

【請求項7】 前記CDインターフェースは、デジタルビデオディスクによって生成されたシリアル圧縮データをパラレルデータに変換するためのシリアル-パラレル変換器を更に備えることを特徴とする請求項5の適合選択法。

30 【請求項8】 前記入力インターフェースは、デジタル放送局によって生成されたMPEG圧縮データを受信するためのデジタル放送受信インターフェースであることを特徴とする請求項3の適合選択法。

【請求項9】 前記音声処理器は、前記音声解凍出力信号としてPCM出力を生成するPCM処理器であることを特徴とする請求項3の適合選択法。

【請求項10】 前記映像処理器は、前記映像解凍出力信号としてNTSC出力を生成するNTSC処理器であることを特徴とする請求項3の適合選択法。

40 【請求項11】 前記映像処理器は、前記映像解凍出力信号としてPAL出力を生成するPAL処理器であることを特徴とする請求項3の適合選択法。

【請求項12】 前記映像処理器は、前記映像解凍出力信号としてVGA出力を生成するVGA処理器であることを特徴とする請求項3の適合選択法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般にMPEG (Motion Picture Experts Group) 回路におけるメモリアクセス優先度制御に関し、特にMPEG処理器におけるメモリアクセス優先度制御の適合選択法に関する。更に特定すれば、本発明は、システム資源のテークオー

バーを減少させることによって解凍性能の改善を図るために、MPEG処理器におけるメモリアクセス優先度をダイナミック制御するための適合選択法に関する。

【0002】

【従来の技術】デジタル信号処理技術、素材工学、レーザ技術を含む分野の進歩の結果として、デジタル形態の音声および映像信号の記憶と検索は、高品質の音響および動画の複製産業にとって自然な選択となっている。娯楽番組の放送では、数10年前の技術に基づいた長年のアナログ形態から離れて、デジタル形態に向かう同様の傾向がある。

【0003】テレビジョン受像機やラジオ受信機のように消費者側にあるアナログ受信設備の装備ベースが膨大である故に、購買者側に対する信号送信の最後の部分を依然としてアナログ体系で実行しているが、番組信号が操作されたり処理されるときには、それが記憶/検索の処理であっても送信の処理であっても、デジタル体系の使用に常に人気がある。例えば、地上局に対する衛星放送デジタル信号は、その番組信号をアナログ形態に変換してからケーブルネットワークを介して購買者の家に中継される。広く期待されている高画質TV(HDTV)に関連する全デジタル放送システムのようなシステムに対しては、複数の基準が提案されている。アナログ処理体系から離れてデジタル処理体系に移行する上述した傾向は、現在利用できるものに基づくことにより、音声および映像信号に対するデジタル記憶および検索技術がアナログのものに比べて優れており、しかもより良い結果を提供するからである。デジタル処理は、人間の聴覚および視覚双方の生理学的受容能力が要求する優れた品質の音響と映像を複製する、コスト対効果比に優れた実際的に唯一の手段を提供する。

【0004】種々のデジタル信号圧縮/解凍体系の中で、MPEG基準が、MPEG-IであってもMPEG-IIであっても、マルチメディア産業界で最も信頼され、且つ広く受け入れられるものとして出現してきた。信号解凍において、即ち複製側では、他の多くと同様に、圧縮音声および映像データを含む信号を提供する供給源から番組再生用のデータ検索を実行するために、MPEG体系はデジタル信号処理(DSP)回路要素を使用することに依存する。再生装置におけるMPEG処理回路に対する圧縮データの供給源は、例えばビデオCD(VCD)やデジタルビデオディスク(DVD)を含むデータ記憶用の人気のあるコンパクトディスク(CD)ファミリーの最新のものである。MPEG処理回路はまた、デジタル放送局からの圧縮データ信号源を受信する。

【0005】MPEG基準を使用するマルチメディアの適用において、信号源から取り込んだ圧縮データを使用する音声および映像の複製の実行には、MPEG処理器のような公用に供されるデジタル電子ハードウェアを使

用する必要がある。これらのMPEG処理器は、デジタル信号処理器とMPEG解凍動作実行用のファームウェア体系を実行するマイクロプロセッサの回りに組み立てられたデジタル回路要素を使用して構成される。メモリ資源もまた、MPEG解凍を実行する過程で使用される。事実、MPEG処理器は、マルチメディアデータが番組再生用に解凍されるとき、メモリサブシステムの使用に強く依存する。

【0006】ところが、音声および映像信号解凍のMPEG基準を実行するデジタル電子回路における従来のハードウェアモジュールは、小型で自立型のファームウェアシステムでは、固定式のメモリアクセス優先度を使用している。その様な従来のMPEGシステムでは、システム資源の利用は、システムのCPU、DSP(デジタル信号処理器)、メモリ、および支援用論理回路の全てに結合されたデータバスに与えられたバンド幅の利点を全て享受できるようには最適化できない。デジタル処理分野の当業者には周知のように、デジタルシステム内の資源のアンバランスな使用は、システム全体の能力の浪費に直接つながる。システム内の多くの構成部品についての性能の可能性の増加は必要不可欠ことになる。その様な性能の増加は、資源をバランス良く使用しているシステムと同等のレベルのシステム処理能力を達成するために必要である。換言すれば、全ての構成モジュールの中で資源(バスのバンド幅を含む)をアンバランスに導入しているMPEGシステムは、資源をバランス良く使用しているシステムと比較したとき、より性能の高いCPU、DSP、または他の回路のいずれかを使用することが要求される。

【0007】特に、MPEG解凍動作の場合には、MPEG処理器内の全ての機能モジュール間でメモリアクセスの優先度が固定されていると、CPUがファームウェアループを無限に繰り返す状態にトラップされたときに、メモリバスのバンド幅を無駄に消費する現象が生じる。一方、MPEG処理器内のモジュールがシステムバスを介して資源にアクセスしようするとき、前記バスが占有されている状況がしばしばある。この場合、要求したモジュールは、待機状態にされねばならない。この結果、CPUがそのポーリングを実行するように制御するために、システムは相当の時間を費やす。一方、システムのDSPセクションでは、メモリシステム内のデータにアクセスするために、バスへのアクセス権を獲得しようとする順路に沿って渋滞が起こる。

【0008】

【発明が解決しようとする課題】それ故、本発明の1つの目的は、メモリバスのバンド幅のバランスのとれた使用を達成するために、MPEG処理器内のメモリアクセスの優先度制御を行う適合選択法を提供することにある。

【0009】本発明の他の目的は、MPEG解凍性能全

体の速度を改善するために、メモリバスのバンド幅のバランスのとれた使用を達成するMPEG処理器内のメモリアクセスの優先度制御を行う適合選択法を提供することにある。

【0010】本発明の更に他の目的は、MPEG解凍性能全体の速度を改善するために、システムバスへのアクセス権のダイナミックな優先度調整によって、メモリバスのバンド幅のバランスのとれた使用を達成するMPEG処理器内のメモリアクセスの優先度制御を行う適合選択法を提供することにある。

【0011】

【課題を解決するための手段】本発明は、MPEG処理器内のメモリアクセスの優先度制御をする適合選択法を提供することによって、上述した目的を達成する。前記処理器は、MPEG圧縮データから音声圧縮データおよび映像圧縮データを解剖するためのCPUを含む複数の機能モジュールを備える。そしてメモリ制御器は、前記データバスを介して前記メモリをアクセスする全てのモジュールのそれぞれのアクセス優先度を仲裁するために使用される。前記データバスに対する前記中央処理ユニットのアクセス優先度は、前記中央処理ユニットがMPEG圧縮データを解剖する必要があるときおよび前記音声圧縮データを生成する初期復号を行うとき以外は、相対的に低く維持される。それ故、データバスのバンド幅の使用はバランスされ、これによりシステム全体の性能が増大する。

【0012】本発明の他の目的、形態および利点は、以下に示す好ましいが限定されるものではない実施例の詳細な説明によって明らかになる。この説明は添付した図面を伴って行われる。

【0013】

【発明の実施の形態】図1を参照すると、典型的なMPEG処理器の内部構成を図示するブロック図が示されている。その様なMPEG処理器のハードウェア回路構成と一般的な動作が、本発明の説明のために検証される。

【0014】前記ブロック図に図示されているように、参照符号100で示されるMPEG処理器は、データバスと複数の制御信号線のネットワークによって相互に内部結合された複数の機能モジュールを有している。MPEG処理器100は、MPEG圧縮基準に従った入力圧縮データを一端で受信し、そして処理後に解凍音声および映像の番組信号を他端に生成する。

【0015】図示の実施例では、MPEG処理器100は、CD互換デバイス、例えばVCDまたはDVDから一連の圧縮データを受信し、そして音声出力としてPCM信号を、また映像出力としてNTSC信号を生成する。周知のように、MPEG処理器100の入力はまた、MPEG基準に従ったデジタル信号を送信する放送局のようなマルチメディア信号源からも到来する。一方、MPEG処理器100で生成された映像出力信号

は、PAL信号か、またはパーソナルコンピュータ産業で一般的な、例えば標準VGAフォーマットの形態をとる。この映像出力信号は、更に処理および表示するために適切な回路に中継される。

【0016】図1のハードウェア構成において、MPEG処理器100は、参照符号400で示されるメモリシステムと協同して、受信したMPEGデータを解凍する動作をする。この実施例では、MPEG解凍体系の実行に必要なメモリシステム400内のメモリブロックは、MPEG処理器100から物理的に独立している。MPEG処理器100は、両者の間を結合するデータバスを介してメモリ400にアクセスする。全ての当業者が気づくように、MPEG処理器の外部に物理的に配置されたメモリブロックの使用は、必ずしも必要なことではない。MPEG処理器の内部に作業用メモリブロックを含ませることもまた可能である。図1に図示された実施例の場合、1つの特殊な状況において、MPEG処理器100は、パーソナルコンピュータの延長バスに装着されたVCD（またはDVD）ドライブサブシステムに結合する対象になり得る。この特殊な配置は、ホストコンピュータシステム内のアドレス指定可能なメモリ空間の選定されたメモリ部分を、作業用メモリ領域として取得する。

【0017】図1に示されたMPEG処理器100は、処理器それ自体とMPEG圧縮信号源との間のインターフェースとして使用されるCDインターフェースモジュール110を備える。この信号源は、この実施例で説明されるように、CD互換デバイスとしてのVCDまたはDVDである。通常の場合では、CDインターフェース110は、シリアル転送されるMPEG圧縮フォーマットに調整されたデータ信号を受信する。これは、標準CD互換デバイスが、他の多くの磁気媒体ベースのドライブと同様に、一連のシングルビットの形態でメディアの記録表面にストアされたデータをアクセスするからである。図面には示されていないが、CDインターフェース110はかくして、処理器の内部回路におけるMPEG基準の規格に沿った後続処理のために、受信したシリアルデータをパラレルデータに変換するシリアル・パラレル変換回路を含む必要がある。かくして処理された入力データは、それからFIFO（ファーストイン・ファーストアウト）112に一時蓄積され、そして更なる処理のために処理器100内の後続する回路モジュールに送られる。

【0018】CDインターフェース110は、データバスMEM-BUSを介して、MPEG処理器100の残りの回路に結合される。基本的に、図示の実施例では、データバスMEM-BUS上に存在する処理器用作業用メモリとなるメモリシステム400は、図面内で判別できる。MPEG処理器100の主機能回路モジュールはまた、それらが動作するときメモリシステム400を

アクセスできるように、このデータバス上に存在する。図面で、機能モジュールに導入され、またそこから導出されるバスセグメントの双方向の印は、必要があれば、データが双方向に伝送されることを示している。

【0019】MPEG処理器100の機能モジュールには、システムに対し入力インターフェースを提供するCDインターフェース110の他に、CPU120、MPEG音声デコーダ130、PCM処理器132、MPEG映像デコーダ140、映像処理器142、およびメモリ制御器150が含まれる。上述したように、これらのモジュールはデータバスMEM-BUS上に存在し、MPEG処理器100が動作し、CDインターフェース110を介して受信した外部ソース上のMPEGデータから音声および映像出力を生成するときに、メモリシステム400にアクセスすることを許容する。

【0020】CPU120は、MPEGデータ解凍処理においてMPEG処理器100内の機能モジュールの動作と協同してファームウェアルーチンを実行するマイクロプロセッサまたはマイクロコントローラである。一旦このルーチンが開始されると、CPU120は、予めプログラムされた優先度体系内の全ての機能モジュールと協同する。この優先度体系は、メモリ制御器150の制御下で要求された時に、各モジュールがメモリ資源にアクセスする事を許容するものである。メモリ制御器150が、データバスMEM-BUSを介してどのようなモジュールに対しても、優先度体系に基づくメモリ資源400へのアクセス権を許可すると、そのモジュール、即ちCDインターフェース110、MPEG音声デコーダ130、PCM処理器132、MPEG映像デコーダ140、映像処理器142およびCPU120の中から選択された1つが、独立してメモリ資源400にアクセスすることが許容される。

【0021】当業者には周知のように、共通のデータバス上に存在する複数のデバイスは、共有する目標メモリ資源を同時にアクセスすることが許容される。これは、優先度決定の設定体系下に導入されたデータバスMEM-BUSへのアクセス権に係る競合の過程である。従来のMPEG処理器では、この体系は固定された方法である。その様な従来の優先度体系は、全ての機能モジュールの要求状態を監視し、そしてファームウェアルーチンに記憶されている優先度法に基づくメモリアクセス用のデータバスへのアクセス権を許容する制御要素、図1のハードウェア構成の場合にはメモリ制御器150を必要とする。

【0022】図1に示されたハードウェア構成例の場合には、MPEG処理器100内の複数の機能モジュールはメモリ制御器150の仲裁の下に協同し、規則正しい手法でデータバスMEM-BUS上のアクセスを獲得する。MPEG処理器100内の機能モジュールのそれぞれは、データバスMEM-BUSを介してメモリ資源4

00に接続される以外は、メモリ制御器150に接続されたハンドシェイク式の制御信号線を追加的に備えている。これらの制御線は、メモリ資源への各モジュールによるアクセス動作の制御を容易にする。

【0023】そして、MPEG処理器100内のCPU120はまた、MPEG圧縮データを、音声、映像、およびMPEG基準を使用した圧縮データからなる他の支援用データセグメントに解剖する責任を持つ。図1に示されたハードウェアの例では、上述したように、CDインターフェース110は、外部ソースから一連の圧縮データのシリアルビットを受信し、そしてその受信MPEG圧縮データをメモリ資源400のCD用FIFO422にストアする。上述したように、この過程は、メモリ制御器150による協同を必要とする。それから、ファームウェアルーチンの制御の下で、CPU120はまたCD用FIFO422から検索したデータの解剖を実行し、生成した音声および映像圧縮信号を音声バッファ412と映像バッファ414にそれぞれストアする。

【0024】一方、MPEG音声デコーダ130とMPEG映像デコーダ140は、対応する音声および映像データを圧縮されていない形態に復号するために、実際にそれぞれ対応するデータを分解する音声および映像DSPの役割を実質的に果たすものである。周知のように、これらの動作は復号アルゴリズムを使用する。

【0025】例えば、MPEG映像デコーダ140がメモリ資源400にアクセス要求するときは、それはメモリ制御器150への制御線VD-MEM上にリクエスト信号を出す。メモリ制御器150は、上記リクエストを受信すると、所定のメモリアクセス優先度体系に基づいて仲裁し、そしてメモリ制御器150による仲裁結果がデータバスMEM-BUSへのアクセスを許容するものであれば、MPEG映像デコーダ140はデータバスMEM-BUSを介してメモリ資源400へのメモリアクセスを開始できる。

【0026】MPEG映像デコーダ140はそれから、例えば、選定した位置、即ちメモリ資源400の映像バッファ414にストアされているデータを検索し、CPU120によってそれ以前に解剖され、そして映像バッファ414に書き込まれていた圧縮映像データの処理を実行する。一方、MPEG映像デコーダ140はまた、例えば、その生成したデータを選定した位置、即ちメモリ資源400内のフレームバッファ432にストアする。そして、フレームバッファ432にストアされたデータは、メモリ制御器150の仲裁を含む手法と同様に、後に映像処理器142によって検索される。映像処理器142はそれから、生成した結果を映像出力、即ち図1の場合にはNTSC信号として出力する。周知のように、映像処理器142は、他の場合にはPAL信号を出力するものになり得る。

【0027】かくして、図1に概要を示したハードウェア

ア構成を有するMPEG処理器100において、CPU 120は、MPEG圧縮データの解剖と、音声および映像の初期復号を効果的な手法で実行する必要がある。換言すれば、CPU120は、その様な解剖と初期復号をできるだけ速く終了するために、データベースに対する高い優先度のアクセス権を割り当てられねばならない。上述したように、従来のMPEG処理器は、固定優先度体系の下でこの動作を容易にしている。その様な固定優先度法では、一旦CPU120がプログラムループのステップに従事すると、MPEG処理器100内の全ての機能モジュールはそれらの対応する機能呼び出し動作を実行することが許容されない。この固定優先度体系は、少なくとも1つの大きな問題点を受ける。因みに、CPU120それ自体はまた、それが自らのルーチンを実行してメモリ資源400にアクセスするとき、データベースMEM-BUSのバンド幅を消費する。それ故、MPEG処理器100内のどの機能モジュールがデータベースMEM-BUSへのアクセス要求を出しているかを見るためにスキャンする一時的なプログラムループに、CPU120がトラップされる事態がしばしば発生する。このトラップされている期間は、データベースMEM-BUSがCPU120によってロックアップされているため、全ての機能モジュールはそれらの対応する機能を実行することが許容されなくなる。かくして、その状況は、MPEG処理器100が実際にMPEGデータの解凍を実行するより長い時間をループ内で消費するという結果に終わる。それ故、固定優先度体系を使用する従来のMPEG処理器全体の性能は、全く不十分なものになる。

【0028】例えば、CDドライブ（初期のCDファミリーから開発された最新のメンバーであるVCDやDVDを含む）の出力データがシリアル形態であるため、メモリ資源400内の空のCD用FIFO422がMPEG処理器100全体の内部動作のボトルネックを形成するようになる。従来のMPEG処理器によって採用される固定優先度法は、真にデータベースMEM-BUSにアクセスして自らの機能を実行する必要がある他の機能モジュールを許容する柔軟性がないため、そのような状況でボトルネックが形成される。MPEG処理器内の全ての機能モジュールに等しく同じ優先度が割られるように、それらが無限のサイクルで回転する必要がある。このことは、それらが1つのループ内を回転し、そしてそれぞれが参加する前に、同じステップのシーケンスに追従する必要がある。

【0029】図2は、無限ループで動作する従来のMPEG処理器のファームウェアの概要を示すフローチャートである。この従来のルーチンは固定優先度法に基づいており、そして外部から受信したMPEG圧縮データの解凍動作の実行を制御するために使用されるものである。図2に示されるように、図1のハードウェア構成内のCPU120によって実行されるこの従来のファーム

ウェアルーチンは、一旦ステップ200でスタートすると、ステップ220からは循環する連続したループである。特に、ルーチンがステップ200でスタートするとき、MPEG処理器100は、全ての実行される機能に対する優先度体系の内部条件を、MPEG解凍を実行する前記処理器によって設定する。

【0030】図2に示す従来の無限ループでは、MPEG処理器100に含まれる全ての機能モジュール、即ちCDインターフェース110、MPEG音声デコーダ130、PCM処理器132、MPEG映像デコーダ140およびCPU120は全て、データベースMEM-BUSにアクセスする必要があるときは、メモリ制御器150による仲裁を受ける。メモリアクセスの優先度がステップ210において設定され、そしてこの優先度を変更する他のステップがないため、解凍データの解剖や音声データおよび映像データの復号という基本的な機能は、同一のそして変更されることのない優先度水準を甘んじて受ける必要がある。

【0031】図2において、循環ルーチンは、最初にステップ220で音声復号機能が必要であるか否かを判定する。この決定作成ステップのために、CPU120は音声復号を実行する必要があるか否かを決定する。循環ルーチンはステップ222にジャンプする。ここでは無限ルーチンがループから一時的に分岐し、サブルーチンで実行される機能呼出、即ちこのステップブロックとして認識される機能呼出Aを実行する。この呼び出されたサブルーチンでは、CPU120はメモリ資源400にストアされている圧縮音声データに対応するデータを復号する。復号の後で、CPU120は復号された圧縮音声データを前記メモリ資源に出力する。この後、プログラムステップ222で概要が示されるプログラムサブルーチンは終了され、そしてプログラム制御はメイン循環ループに復帰する。換言すれば、ループは更に処理するためにステップ230に遷移する。

【0032】もし、これとは逆に、ステップ220でCPU120が初期音声復号の必要がないと判定したら、図2のルーチンはステップ230に進む。

【0033】同様に、ステップ230では、CPU120は、MPEG処理器100のCDインターフェースによって外部ソースから抽出されたCDデータやMPEGビットストリームの解剖が必要であるか否かを判定する。適切なフラグを立てることによってその機能が要求されているときは、CPU120は再びメインプログラムルーチンから分岐して、プログラムステップ232の機能呼出Bとして概要が示される一連の動作を実行するために協同する。これらはCPU120がCDインターフェース110を介して得たCDデータを解剖することを含んでいる。MPEG解剖動作の結果として得られた音声圧縮データに対応するデータは、それからメモリ資源400に出力される。解剖された映像データはまた、

11

初期映像復号に続いて、CPU120によって復号され、その結果はメモリ400の映像バッファ414に出力される。それから、プログラム遷移は再びメインループに復帰し、ステップ240進むように継続する。

【0034】もしシステムがステップ230で機能呼出Bへのプログラム分岐が必要ないと判断したら、メイン循環ループはステップ240へ進む。このステップ240でルーチンは、CPU120によって実行される機能呼出Cとして分岐ステップ242に概要が示される更なるMPEG解凍機能が必要であるか否かを判断する。もしイエスであれば、CPU120はプログラム分岐の推進に協同し、それからメインループに復帰する。もし判定結果が否定的なものであれば、ループはメインプログラム循環を維持し、ルーチン循環が再度繰り返されるステップ220に戻る。

【0035】従来のMPEG処理器用の図2のファームウェアルーチンでは、メイン循環ルーチンから分岐した機能呼出サブルーチンのサービス、即ちステップ222、232および242に概要が示された動作は、固定的な参加の体系として設定されている。上述したように、前記サービスモジュールのいくつかが実際に実行される前に、機能呼出A、BおよびCが複数回繰り返されなければならないとすれば、メインプログラムループを通しての循環には、相当な時間の浪費がある。

【0036】図3のフローチャートに示す本発明の実施例は、MPEG処理器ファームウェアルーチンの実効的なサービス任務サイクルを改善し、これによりMPEG解凍全体の実効性を推進するダイナミックサービス優先度割当体系を有している。図3に示すように、そのフローチャートは、本発明の好ましい実施例に伴うMPEG処理器用のファームウェアルーチンを示している。このルーチンは、MPEG基準に従う圧縮データに対し実行される解凍の実行を制御するために使用される。

【0037】図3のフローチャートに概要が示されるファームウェアルーチンの詳細な説明のために、図1に示されたMPEG処理器100の使用が依然として考慮される。図示されているように、図1のハードウェア構成におけるCPU120によって実行される例示のファームウェアルーチンはまた、一旦ステップ300でスタートすると、ステップ320から循環する連続したプログラムループの周りに構成される。特に、ステップ300でルーチンがスタートするときに、MPEG処理器100は、MPEG解凍の実行過程で前記処理器によって実行される全ての機能に対するメモリアクセスの優先度体系用初期条件を確立する。しかしながら、この優先度条件の設定は単に初期的な設定であり、MPEG処理器100の動作がそのタスクを実行するときに、そのパラメータはダイナミックに調整される。

【0038】図3から明らかなように、循環するメインルーチンは最初にステップ320で音声復号機能が必要

12

であるか否かを判定する。この判断作成ステップに対し、CPU120が初期音声復号の必要があると判定したときは、ファームウェアルーチンはステップ322にジャンプし、そこでCPU120のデータバスMEM-BUSに対するアクセス優先度が上昇される。優先度の上昇は、既にルーチンがスタートしている場合には、ステップ310で設定された最初の水準に対して行われる。それからルーチンはステップ324に進み、そこで機能呼出Aで認識される1つの機能呼出が、対応する機能サブルーチンを実行することで遂行される。上述した従来例に対する場合と同様に、CPU120はメモリ資源400にストアされている音声圧縮データに対応したデータを復号し、それからその結果を再びメモリ資源400にストアする。これは、メモリ制御器150の制御下にあるデータバスMEM-BUSへのアクセスを支配するCPU120によって容易にされる。この後、プログラムステップ324に概要が示されるプログラムサブルーチンを完了することができ、そしてプログラム制御はステップ326に遷移する。ステップ326では、データバスMEM-BUSを介してメモリ資源400にアクセスするCPU120に割り当てられた優先度水準は低下される。この時、データバスを介してサービス要求をするためのCPU優先度を、初期設定したものより低い水準にまで低下させることができる。

【0039】一方、CPU120がステップ320で初期音声復号が不要であると判定したら、図3のルーチンは直接ステップ330に進む。

【0040】ステップ330では、ステップ320での手法と同様に、CPU120はその他の機能サブルーチンに対するサービスが要求されているか否かを決定する。例えば、ステップ330は、MPEG処理器100のCDインターフェース110によって外部ソースから抽出されたCDデータやMPEGビットストリームが解剖されるべきものか否かを判定する。もし適切なフラグを立てることによってこの機能が要求されているときは、ルーチンはステップ332に進んでデータバスMEM-BUSへアクセスするCPU120の優先度水準を上昇させる。この優先度上昇は、ルーチンが初期にスタートしたときステップ310で設定された最初の水準に対して行われる。それから、ルーチンはステップ334に進み、そこで機能呼出Bで認識される1つの機能呼出が対応する機能サブルーチンを実行することで遂行される。再び、上述した従来のルーチンに対する場合と同様に、MPEGデータ解凍を実行する過程で必要になる機能動作がこのサブルーチンで実行される。例えば、この呼び出されたサブルーチンの動作には、CDインターフェース110を介して得られたCDデータをCPU120で解剖することを含む。MPEGビットストリームもまたシステム水準の解剖の対象になる。MPEG解剖動作の結果として得られる音声圧縮データに対応した

13

データは、それからメモリ400に出力される。映像データが解剖され、それからCPU120によって初期映像復号が実行され、そして得られたデータがメモリ400内の映像バッファ414に出力される。それから、プログラム制御はステップ336に移され、ここでデータバスMEM-BUSを介してメモリ資源400にアクセスするCPUの優先度は通常に低下させられる。このステップの後に、ルーチンはステップ340に進む。

【0041】しかし、システムがサービス機能呼出へのプログラム分岐が必要ないと決定したときは、メイン循環ループはステップ340に進む。ルーチンは、CPU120によって実行される機能呼出Cとして概要が示される分岐ステップ342の更なるMPEG解凍機能が必要であるか否かを、ステップ340で決定する。もしイエスであれば、CPU120は分岐することに協同し、それからメインループに復帰する。もし決定が否定的であれば、ループはメインプログラムサイクルにとどまり、そしてルーチンサイクルが再び繰り返されるステップ320に戻る。

【0042】

【発明の効果】MPEG処理器に対する本発明の好ましい実施例を示す図3のファームウェアルーチンでは、メイン循環ループから分岐する機能呼出サブルーチンに対するサービス、即ちステップ324、334および342に概要が示された動作は、参加優先度のダイナミックな体系に設定されている。データバスMEM-BUSを介してメモリ資源400にアクセスするCPUの優先度は、必要であるときのみ通常より高い水準に上昇され、CPU120がデータバスの制御を獲得する必要のない他の全ての期間は、そのアクセス優先度は通常より低く

14

に占有する状況が生起することを防止する。即ち、CPU120を指摘する優先度リストの回転という単純な理由によって、他の機能モジュールが制御を獲得しようとしている間にデータバスが不要に占有されることがなくなる。図3に示されたフローチャートにおいて、ファームウェアルーチンのメイン循環ループが、データバスのアクセス権を許可する必要があるときに、CPU120を指摘していても、MPEG処理器100の他の機能モジュールがメモリ資源400へのアクセスを獲得することに対する障害は存在しない。これは、CPUの優先度が相対的に低い水準に維持されているためである。かくして、MPEG処理器の解凍性能は、従来の方法のそれに比べて実質的に改善される。

【0043】本発明は例示する手法により、そして好ましい実施例に関して説明されたが、本発明は開示した実施例に限定されるものではないことが理解される。逆に、本発明は添付した請求の範囲の精神内に含まれる種々の変形と同様の構成をカバーすることを意図している。本発明の請求の範囲は、そのような変形と同様の構成を全て包含するように最も広く解釈されるべきである。

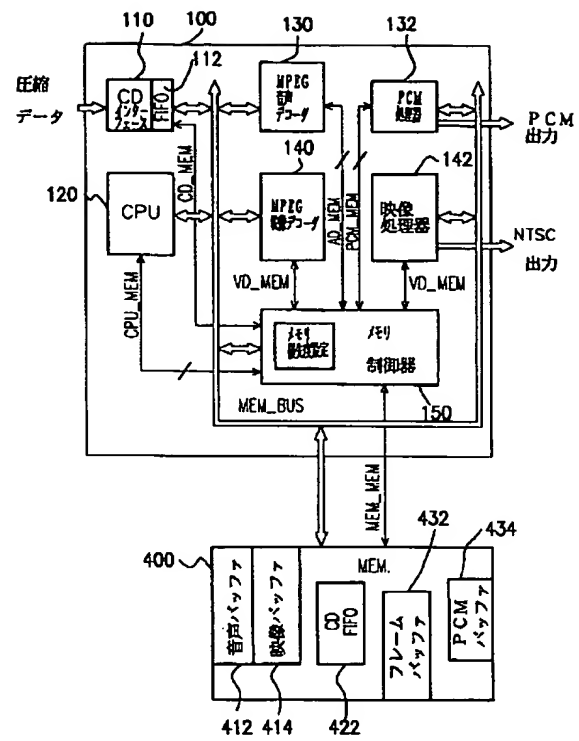
【図面の簡単な説明】

【図1】 MPEG処理器の内部構成を示すブロック図である。

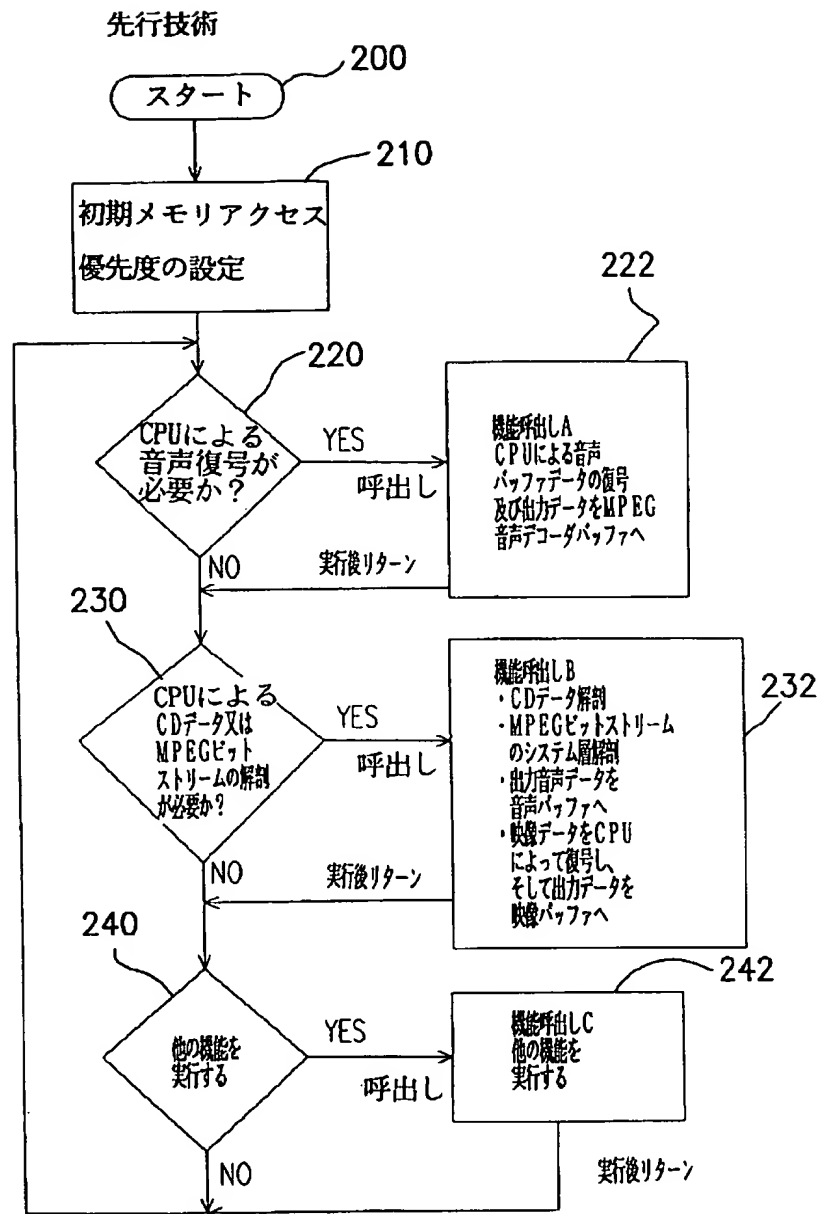
【図2】 固定された優先度体系で解凍動作の実行制御に使用される従来のMPEG処理器のファームウェアルーチンを示すフローチャートである。

【図3】 適合選択優先度体系で解凍動作の実行制御に使用されるMPEG処理器の本発明の好ましい実施例のよるファームウェアルーチンを示すフローチャートである。

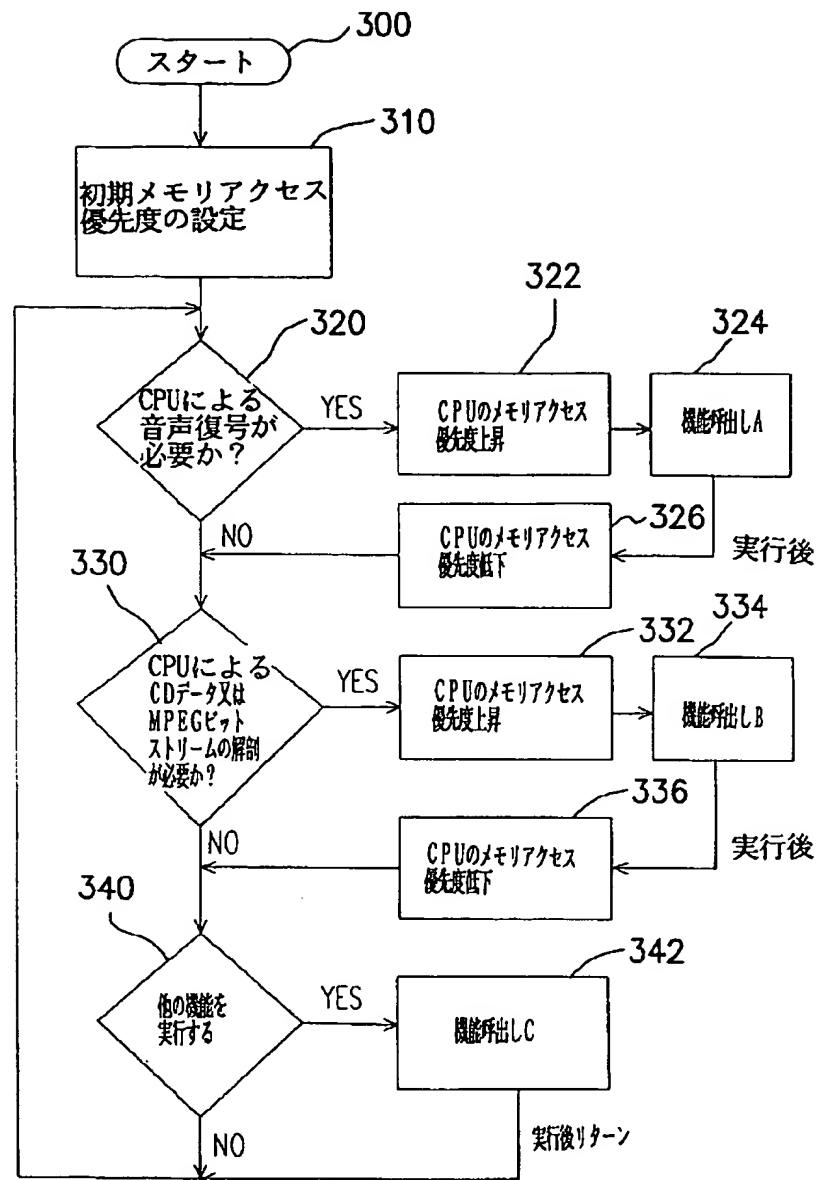
【図1】



【図2】



【図3】



フロントページの続き

(51)Int. Cl.⁶

H04N 7/03
7/035
7/08
7/081

識別記号

F I

【外国語明細書】

1 . T i t l e o f I n v e n t i o n

ADAPTIVE-SELECTION METHOD FOR MEMORY ACCESS PRIORITY CONTROL IN MPEG PROCESSOR

2 . C l a i m s

1. An adaptive-selection method for memory access priority control in MPEG processor, said processor comprising:

an input interface for receiving compressed data and generating MPEG compressed data;

a central processing unit for parsing audio and video compressed data from the MPEG compressed data;

an audio decoder and a video decoder for decoding audio and video data from the audio and video compressed data respectively;

an audio processor and a video processor for generating audio and video decompressed output signals from the audio and video data respectively; and

a memory controller arbitrating the access right over a data bus storing the MPEG data, audio and video compressed data, audio and video data in a memory;

wherein the input interface, the central processing unit, the audio and video decoders, the audio and video processors and the memory controller are linked together over the data bus for communicating data therebetween; the adaptive-selection method is an endless routine and comprising the steps of:

raising the access priority of the central processing unit to the data bus if the central processing unit needs to perform initial audio decoding, and lowering the raised access priority after the initial decoding of the audio data; and

raising the access priority of the central processing unit to the data bus if the central processing unit needs to parse the audio and video compressed data, implementing the parsing of the audio and video compressed data by the central processing unit, and lowering the raised access priority after the parsing.

2. The adaptive-selection method of claim 1, wherein the step of raising the access priority of the central processing unit to the data bus if the central

processing unit needs to parse MPEG compressed data and then implementing the initial audio decoding further comprises:

the central processing unit retrieving the MPEG compressed data stored in the memory via the data bus for parsing to generate the data corresponding to the audio compressed data and the data corresponding to the video compressed data;

storing the data corresponding to the audio compressed data in the memory via the data bus;

initial decoding the data corresponding to the video compressed data to obtain the video compressed data; and

storing the video compressed data in the memory via the data bus.

3. The adaptive-selection method of claim 2, wherein the step of raising the access priority of the central processing unit to the data bus if the central processing unit needs to parse MPEG compressed data and then implementing the initial audio decoding further comprises:

the central processing unit retrieving the data corresponding to the audio compressed data stored in the memory via the data bus for initial decoding to generate the audio compressed data; and

storing the audio compressed data in the memory via the data bus.

4. The adaptive-selection method of claim 3, wherein the input interface is a CD interface for receiving MPEG compressed data generated by a Video Compact Disc.

5. The adaptive-selection method of claim 3, wherein the input interface is a CD interface for receiving MPEG compressed data generated by a Digital Video Disc.

6. The adaptive-selection method of claim 4, wherein the CD interface further comprises a serial-to-parallel converter for converting the serial compressed data generated by the Video Compact Disc into parallel data.

7. The adaptive-selection method of claim 5, wherein the CD interface further comprises a serial-to-parallel converter for converting the serial compressed data generated by the Digital Video Disc into parallel data.

8. The adaptive-selection method of claim 3 wherein the input interface is a digital broadcast receiving interface for receiving compressed data generated by a digital broadcast station.

9. The adaptive-selection method of claim 3, wherein the audio processor is a PCM processor for generating PCM output as the audio decompressed output signal.

10. The adaptive-selection method of claim 3, wherein the video processor is an NTSC processor for generating NTSC output as the video decompressed output signal.

11. The adaptive-selection method of claim 3, wherein the video processor is a PAL processor for generating PAL output as the video decompressed output signal.

12. The adaptive-selection method of claim 3, wherein the video processor is a VGA processor for generating VGA output as the video decompressed output signal.

3 . D e t a i l e d E x p l a n a t i o n o f t h e I n v e n t

i o n

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates in general to the memory access priority control in MPEG (Motion Picture Experts Group) circuitry, and in particular to an adaptive-selection method for memory access priority control in an MPEG processor. More particularly, this invention relates to an adaptive-selection method for dynamic control of memory access priority in an MPEG processor for improving decompression performance by reducing unnecessary takeover of the system resource.

Description of the Related Art

As a result of advancements in fields including the digital signal processing technology, material science, as well as laser engineering, storage and retrieval of audio and video signals in digital format have become the selection of nature for high-fidelity sound and motion picture reproduction industry. In the broadcast of entertainment programs, there has been the similar trend of change toward digital format and away from the aging analog format that laid its technological basis several decades ago.

Due to the huge installation base of analog receiver equipment at the consumer end such as television sets and radio receivers, other than the last segment of signal transmission to the subscriber end is still implemented in the analog scheme, there has been the ever popular use of digital format when the program signal is manipulated or processed, either in the process of storage/retrieval or transmission. For example, satellites broadcast digital signals to ground stations, which then convert and relays the program signal in analog to the subscribers' houses via the cable network. There has even been several proposed standards for all-digital broadcast systems such as those incorporated

with the widely expected High-Definition TV (HDTV). The above-mentioned trend of switching away from the analog toward the digital processing scheme is because, based on what are currently available, the technology of digital storage and retrieval for audio and video signals, better than its analog counterpart, provides much better results. Digital processing provides virtually the only means for superior quality of sound and video reproduction in the cost-effective manner that human physical perception capabilities of both hearing and vision can demand.

Among the various digital signal compression/decompression schemes, MPEG standard, either MPEG-I or MPEG-II, emerges as one of the most promising and widely-accepted in the multimedia industry. At the signal decompression, namely, the reproduction end, like many others, the MPEG scheme relies on the use of digital signal processing (DSP) circuit elements to implement data retrieval for the playback of programs from a source that supplies signals containing compressed audio and video data. The source of compressed data for the MPEG processor circuitry in a playback device may be, for example, the latest members in the popular Compact Disc (CD) family of data storage formats that include the Video CD (VCD) or the Digital Video Disc (DVD). Or, the MPEG processor circuitry may also receive its compressed data signal source from a digital broadcast station.

To implement the audio and video signal reproduction utilizing compressed data fetched from signal sources in a multimedia application that employs the MPEG standard, dedicated digital electronic hardware circuitry known as MPEG processors must be used. These MPEG processors may be constructed utilizing digital circuitry elements built around digital signal processors and microprocessors that execute a firmware scheme for the implementation of the MPEG decompression operation. Memory resources are also used in the process of implementing the MPEG decompression. In fact,

MPEG processors rely heavily on the use of memory subsystem as the multimedia data is being decompressed for program playback.

However, conventional hardware modules in the digital electronic circuitry that implements the MPEG standard of audio and video signal decompression employ a fixed memory access priority in a small and self-sustained firmware system. In such conventional MPEG systems, utilization of system resources can not be optimized to take full advantage of the supported bandwidth of the data bus that links the CPU, the DSP (digital signal processor), the memory and the supporting logic circuitry of the system together. As is familiar to persons skilled in the art of digital processing, unbalanced use of resources in a digital system can be translated directly into the waste of overall system power. Increase of the performance capabilities of many of the constituent parts in the system will be necessary. Such performance increase is necessary in order to achieve the same level of system processing capability as that which has well balanced use of resources. In other words, an MPEG system conducting unbalanced usage of resources (including the bus bandwidth) among all the constituent modules would require the use of either more powerful CPU, DSP, or other circuitry when compared to the one that has well-balanced system resource utilization.

In particular, in the case of MPEG decompression operation, if the memory access priority is fixed among all the functional modules in the MPEG processor, there would be the phenomenal wastes of memory bus bandwidth as the CPU is trapped in an endless scanning firmware loop. On the other hand, whenever any module in the MPEG processor requires to access resources over the system bus, it is frequently the situation that the bus is occupied. In this case, the requesting module has to be put to waiting status. The result is that the system spends considerable time for the controlling CPU to execute its polling, while the DSP section of the system jams along the way trying to gain access to the bus for accessing data in the memory subsystem.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to provide an adaptive-selection method for memory access priority control in MPEG processor for achieving more balanced utilization of memory bus bandwidth.

It is another object of the invention to provide an adaptive-selection method for memory access priority control in MPEG processor that achieves more balanced utilization of memory bus bandwidth for improving overall MPEG decompression performance rating.

It is yet another object of the invention to provide an adaptive-selection method for memory access priority control in MPEG processor that achieves more balanced utilization of memory bus bandwidth by dynamic priority adjustment of access rights to the system bus for improving overall MPEG decompression performance rating.

The present invention achieves the above-identified objects by providing an adaptive-selection method for memory access priority control in MPEG processor. The processor has functional modules that include a CPU for parsing the audio compressed data and the video compressed data from the MPEG compressed data, and a memory controller is used to arbitrate the access priority of each of the modules over the data bus via for accessing the memory. The access priority of the CPU to the data bus is maintained at a relatively lower level except when the CPU needs to perform parsing on the MPEG compressed data and implementing the initial decoding of the audio compressed data. The use of data bus bandwidth is therefore balanced among all the system resources thereby increasing the overall system performance.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Refer to FIG. 1, a block diagram is shown depicting the internal configuration of a typical MPEG processor. Hardware circuitry structural configuration and general operation of such an MPEG processor is examined for the purpose of the description of the invention.

As is illustrated in the block diagram, an MPEG processor, generally designated by the reference numeral 100, has a number of functional modules inter-linked together by a data bus and a network of several control signal lines. The MPEG processor 100 receives input compressed data complying to the MPEG compression standard at one end, and produces decompressed audio and video program signals at the other after processing.

In the illustrated example, the MPEG processor 100 receives a string of compressed data from a CD compatible device, which may be a VCD or a DVD, and generates PCM signal as audio output, and NTSC signal as video output. As is well known, input to the MPEG processor 100 may also come from a multimedia signal source such as a broadcast station transmitting digital signals complying to the MPEG standard. On the other hand, the video output signal as

generated by the MPEG processor 100 may be a PAL signal or may also be in the form of, for example, the standard VGA format popular in the personal computer industry. This video output signal can then be relayed to appropriate circuitry for further processing and display.

In the hardware configuration of FIG. 1, the MPEG processor 100 operates to decompress its received MPEG data in cooperation with a memory system, generally identified by the reference numeral 400 in the drawing. In this described example, the memory blocks in the memory system 400 necessary for the implementation of MPEG decompression scheme is physically independent from the MPEG processor 100. The MPEG processor 100 accesses the memory 400 via the data bus connecting between the two. As persons skilled in the art should all be well aware, this use of memory block arrangement physically external to the MPEG processor is not absolutely necessary. The inclusion of working memory blocks internal to the MPEG processor is also possible. In the case of the illustrated example of FIG. 1, in one particular situation, the MPEG processor 100 can be one incorporated in the VCD (or DVD) drive subsystem installed to the expansion bus of a personal computer system. This specific arrangement may take a designated memory segment of the addressable memory space in the host computer system as the working memory area.

The MPEG processor 100 as illustrated in FIG. 1 has a CD interface module 110 that is used as the interface between the processor itself and the MPEG compressed signal source. This signal source can be a CD compatible device of either VCD or DVD, as in this described example. Under normal situations, the CD interface 110 receives data signals prepared in the MPEG compressed format that is transmitted in serial. This is because standard CD-compatible drives, like many other magnetic media-based drives, access data stored on the surface of the storage media thereof in a stream of single bits. Though not shown in the drawing, the CD interface 110 may thus include serial-to-parallel conversion circuitry that converts the received serial data into parallel

for subsequent processing in the processor internal circuitry as per the specification of the MPEG standard. The input data thus processed is then buffered in a FIFO (first-in first-out) 112 and can then be cranked out to the subsequent circuitry module in the processor 100 for further processing.

The CD interface 110 is linked to the rest of the circuitry of the MPEG processor 100 via a data bus MEM_BUS. Essentially, in the depicted example, the memory system 400 that serves as the working memory for the processor reside on the data bus MEM_BUS, as is seen in the drawing. The major functional circuitry modules of the MPEG processor 100 also reside on this data bus so as to be able to access the system memory 400 as they operate. The bi-directional indication of the bus segments leading in and out of the functional modules in the drawing schematically points out the fact that data are transmitted bi-directionally, as the need arises.

The functional modules in the MPEG processor 100, other than the CD interface 110 that provides an input interface to the system, include a CPU 120, an MPEG audio decoder 130, a PCM processor 132, an MPEG video decoder 140, a video processor 142, and a memory controller 150. As mentioned, these modules reside on the data bus MEM_BUS, allowing to access the system memory 400 as the MPEG processor 100 operates to generate audio and video outputs from the MPEG data received over the external source via the CD interface 110.

The CPU 120 may be a microprocessor or microcontroller that executes a firmware routine for coordinating the operation of the functional modules in the MPEG processor 100 in the process of MPEG data decompression. Once the routine is started, the CPU 120 coordinates all the functional modules in a pre-programmed priority scheme that allows each of the modules to access the memory resource when required under control of the memory controller 150. When the memory controller 150 grants the right of access to the memory resource 400 via the data bus MEM_BUS based on the prioritized scheme to any

module, the module, namely any selected one among the CD interface 110, the MPEG audio decoder 130, the PCM processor 132, the MPEG video decoder 140, the video processor 142 and the CPU 120, is then allowed to access the memory resource 400 independently.

As persons skilled in the art should all be familiar with, multiple devices residing on a common data bus is allowed to access the shared target memory resource one at a time. This is a process of competition over the access right to the data bus MEM_BUS conducted under the set scheme of priority determination. In the conventional MPEG processors, this scheme is a fixed method. Such prior-art priority scheme requires the controlling element, the memory controller 150 in the case of the discussed hardware configuration of FIG. 1, to monitor the requesting status of all the functional modules, and grants the access right to the data bus for memory access based on the priority method embedded in the firmware routine.

In the case of the depicted hardware configuration example of FIG. 1, the functional modules in the MPEG processor 100 are coordinated under arbitration by the memory controller 150 to gain access over the data bus MEM_BUS in an orderly manner. It should be pointed out that each of the functional modules in the MPEG processor 100, except being tied to the memory resource 400 via the data bus MEM_BUS, is also equipped with additional handshaking control signal lines connected to the memory controller 150. These control lines facilitate the control of each of the modules' access operation to the memory resource.

And, the CPU 120 in the MPEG processor 100 is also responsible for the parsing of the MPEG compressed data into the audio, video and other supporting data segments constituting the compressed data complying to the MPEG standard. In the depicted hardware example of FIG. 1, the CD interface 110, as described above, receives the serial bit stream of compressed data from the external source, and then stores the received MPEG-compressed data in the CD FIFO 422 of the memory resource 400. As described, this process requires the

coordination by the memory controller 150. Then, under control of the firmware routine, the CPU 120 also implements parsing on the data retrieved from the CD FIFO 422, and then stores the generated audio and video compressed signals in the audio buffer 412 and the video buffer 414 respectively.

On the other end, the MPEG audio decoder 130 and the MPEG video decoder 140 substantially play the role of audio and video DSP that actually crunch their respective data for decoding to obtain the corresponding audio and video data in the uncompressed format. As is well known, these operations involves the use of decoding algorithms.

For example, when the MPEG video decoder 140 requests to access the memory resource 400, it flags the requesting signal over the VD_MEM control lines to the memory controller 150. The memory controller 150, upon receiving the request, arbitrates based on the predetermined memory access priority scheme, and if the arbitration result as achieved by the memory controller 150 is to grant the access to the data bus MEM_BUS, MPEG video decoder 140 may then initiate its memory access in the memory resource 400 via the data bus MEM_BUS. The MPEG video decoder 140 may then, for example, retrieve the data stored in the designated location, the video buffer 414 in the memory resource 400, for performing the processing of the compressed video data which were previously parsed and written in video buffer 414 by the CPU 120. On the other hand, the MPEG video decoder 140 may, for example, also store its generated data to the designated location, the frame buffer 432 in the memory resource 400. And, this data stored in frame buffer 432 may later be retrieved by the video processor 142 in a similar manner involving the arbitration of the memory controller 150. The video processor 142 may then output its generated result as the video output, an NTSC signal in the case of FIG. 1. As is well known, the video processor 142 may then output a PAL signal in another case.

Thus, in the MPEG processor 100 having the hardware configuration outlined in FIG. 1, the CPU 120 is required to perform the parsing of the MPEG

compressed data and the initial audio and video decoding in an efficient manner. In other words, the CPU 120 must be assigned high priority of access right over the data bus to conclude such parsing and initial decoding as fast as possible. As mentioned above, prior art MPEG processors facilitate this operation in a fixed priority scheme. In such fixed priority methods, once the CPU 120 is engaged in the steps of the program loop, all the functional modules in the MPEG processor 100 are not allowed to perform their respective function call operations. This fixed priority scheme suffers at least one major drawback. Namely, CPU 120 itself also consumes bandwidth of the data bus MEM_BUS as it performs its routine and access the memory resource 400. Therefore, there is frequently the case that the CPU 120 is trapped temporarily in the program loop scanning to see if any functional module in the MPEG processor 100 requests for access to the data bus MEM_BUS. During this trapped period, all the functional modules are not allowed to perform their respective functionality as the data bus MEM_BUS has been locked up by the CPU 120. Thus, the situation frequently ends up that the MPEG processor 100 spends more time in looping than actually performing the MPEG data decompression. The overall performance of these prior art MPEG processors employing fixed priority scheme are therefore quite inefficient.

For example, due to the fact that CD drives (including VCD and DVD, the latest members developed out of the very original CD family) outputs data in serial format, it is very likely that an empty CD FIFO 422 in the memory resource 400 constitutes the bottleneck of the internal operations in the entire MPEG processor 100. A bottleneck is formed in such a situations since the fixed priority method adopted by these prior-art MPEG processors does not have the flexibility of allowing other functional modules that really needs to access the data bus MEM_BUS for performing its function. They will have to rotate in the endless cycle as all the functional modules in the MPEG processor are equally assigned the same priority. This requires that each of them be rotated in a loop, and each has to follow the same sequence of steps before can get attended.

FIG. 2 shows a flow chart that outlines the firmware routine of a conventional MPEG processor operating in an endless loop. This prior-art routine is based on a fixed priority method and is used for controlling the implementation of the decompression operation performed over MPEG compressed data received externally. As is illustrated in FIG. 2, this prior-art firmware routine executed by the CPU 120 in the hardware configuration of FIG. 1 is a continuous loop that cycles from step 220 once started at step 200. Specifically, when the routine starts at step 200, the MPEG processor 100 sets up the initial conditions for the priority scheme for all the functions to be performed by the processor in the process of implementing MPEG decompression.

In the prior-art endless routine of FIG. 2, all the functional modules in the MPEG processor 100, including CD interface 112, MPEG audio decoder 130, PCM processor 132, MPEG video decoder 140, video processor 142 and CPU 120, are all arbitrated by the memory controller 150 when the need to access the data bus MEM_BUS arises. Since the memory access priority has been set in step 210, and no further step alters this priority, all the basic functionality of either the compressed data parsing, as well as the audio data and the video data decoding, are allowed to enjoy the same and un-altered priority level.

In FIG. 2, the cycling routine first checks to see if the audio decoding functionality is necessary at step 220. For this decision-making step, CPU 120 determines whether or not it is necessary to perform audio decoding. The routine jumps to step 222, where the endless routine branches out of the loop temporarily to perform a function call by executing a subroutine, namely function call A as identified in this step block. In this called subroutine, CPU 120 decodes the data corresponding to the compressed audio data stored in the memory resource 400. After decode, CPU 120 then outputs the decoded compressed audio data to the memory resource. This is implemented by the CPU 120 via access to the data bus MEM_BUS by the memory controller 150. After this, the program subroutine generally outlined in the program step 222 can be concluded, and

program control may then be returned back to the main cycling loop. In other words, the loop transfers to step 230 for further processing.

If, on the other hand, CPU 120 decides in step 220 that it is not necessary to perform initial audio decode, the routine of FIG. 2 will then proceed to step 230.

At step 230, in a similar manner, CPU 120 determines whether or not the CD data or MPEG bit stream as extracted from the external source by the CD interface 110 of the MPEG processor 100 should be parsed. If the functionality is requested via proper flagging, CPU 120 then branches out of the main program loop again and coordinates to perform a series of operations identified as function call B outlined in the program step 232. These included CPU 120 parsing the CD data obtained via the CD interface 110. The MPEG bit stream is also subject to system level parsing. The data corresponding to the audio compressed data obtained as a result of the MPEG parsing operation is then output to the memory resource 400. The parsed video data is also decoded by the CPU 120, followed by the initial video decoding, and the result is then output to the video buffer 414 in memory 400. Then, program transfer is again returned to the main loop, and continues to proceed at step 240.

If the system decided that the program branch to function call B at step 230 is not necessary, the main cycling loop then advances to step 240. The routine decides at step 240 whether or not further MPEG decompression functionality generally outlined in the branched step 242 as function call C as performed by the CPU 120 is necessary. If yes, the CPU 120 coordinates to proceed the program branching accordingly and then returns to the main loop. If the result of decision is negative, the loop just maintains in the main program cycle and gets back to step 220, where the routine cycle is repeated again.

In the firmware routine of FIG. 2 for the conventional MPEG processor, the services to the function call subroutines branched out of the main cycling loop, namely operations outlined in steps 222, 232 and 242, are arranged in the

fixed scheme of attendance. As mentioned above, there is considerable time wasted in cycling through the main program loop as function calls A, B and C would have to be cycled several times before the service modules of any of them can actually be executed.

An embodiment of the invention as demonstrated in the flow chart of FIG. 3 has a dynamic service priority assignment scheme to improve the effective service duty cycle of the MPEG processor firmware routine, thereby promoting the overall MPEG decompression efficiency. As is illustrated in FIG. 3, the flow chart shows the firmware routine in accordance with the preferred embodiment of the present invention for an MPEG processor. This routine is used for controlling the implementation of a decompression performed over the compressed data complying to the MPEG standard.

For a detailed description of the firmware routine outlined in the flow chart of FIG. 3, still consider the use of an MPEG processor 100 as illustrated in FIG. 1. As is illustrated, this exemplified firmware routine executed by the CPU 120 in the hardware configuration of FIG. 1 is also built around a continuous program loop that cycles from step 320 once started at step 300. Specifically, when the routine starts at step 300, the MPEG processor 100 establishes the initial conditions for the priority scheme of memory access for all the functions to be performed in the processor in the process of implementing the MPEG decompression. Note, however, that this set of priority condition is only the initial setting, parameters of which will be dynamically adjusted as the operation of the MPEG processor 100 performs its tasks.

As seen in FIG. 3, the cycling main routine first checks to see if the audio decoding functionality is necessary at step 320. For this decision-making step, if the CPU 120 determines that there is the need to implement the initial audio decoding, the firmware routine jumps to step 322, where the access priority of the CPU 120 to the data bus MEM_BUS is raised. The priority raise is with respect to the original level as set up in step 310 when the routine was initially

started. Then, the routine proceeds to step 324, where a function call identified as function call A is performed by executing a corresponding function subroutine. Similar as in the case described above for the prior-art routine, CPU 120 decodes the data corresponding to the audio compressed data stored in the memory resource 400, and then stores the result in the memory resource 400 again. This is facilitated by CPU 120 governing the access to the data bus MEM_BUS via control by the memory controller 150. After this, the program subroutine generally outlined in the program step 324 can be concluded and program control may then be transferred to step 326, where the priority level assigned to the CPU 120 for accessing the memory resource 400 via control over the data bus MEM_BUS is lowered. At this moment, the CPU priority for requesting service via the data bus can be lowered to a level lower than the initial setting. Then, the firmware routine transfers to step 330 for subsequent MPEG decompression processing.

On the other hand, if initial audio decoding is decided by the CPU 120 at step 320 to be un-necessary, the routine of FIG. 3 will then proceed directly to step 330.

At step 330, in a manner similar to that at step 320, CPU 120 determines whether or not service to another functional subroutine is demanded. For example, step 330 determines whether or not the CD data or MPEG bit stream as extracted from the external source by the CD interface 110 of the MPEG processor 100 should be parsed. If the functionality is requested via proper flagging, the routine then proceeds to step 332 to raise the priority level for the CPU 120 to access the data bus MEM_BUS. The priority raise is with respect to the original level as set up in step 310 when the routine was initially started. Then, the routine proceeds to step 334, where a function call identified as function call B is performed by executing a corresponding function subroutine. Again, similar as was in the case described above for the prior-art routine, dedicated functional operations required in the process of implementing the

MPEG data decompression can be performed in this subroutine. For example, in this called subroutine, the operations include CPU 120 parsing the CD data obtained via the CD interface 110. The MPEG bit stream is also subject to system level parsing. The data corresponding to the audio compressed data obtained as a result of the MPEG parsing operation is then output to the memory resource 400. The video data is parsed, and then the initial video decoding performed by the CPU 120, and the obtained data is then output to the video buffer 414 in memory 400. Then, the program control is transferred to step 336, where the CPU priority for accessing the memory resource 400 over the data bus MEM_BUS is lowered to normal. After this step, the routine continues at step 340.

If, however, the system decided that the program branch to service function call B at step 330 is not necessary, the main cycling loop then advances to step 340. The routine decides at step 340 whether or not further MPEG decompression functionality generally outlined in the branched step 342 as function call C as performed by the CPU 120 is necessary. If yes, the CPU 120 coordinates to proceed the branching accordingly and then returns to the main loop. If the decision is negative, the loop just maintains in the main program cycle and gets back to step 320, where the routine cycle is repeated again.

In the firmware routine of FIG. 3 depicting a preferred embodiment of the invention for operating the MPEG processor, the services to the function call subroutines branched out of the main cycling loop, namely operations outlined in steps 324, 334 and 342, are arranged in a dynamic scheme of attendance priority. The CPU priority to access the memory resource 400 via the data bus MEM_BUS is only raised to a level higher than normal when necessary, and during all other time periods when it is unnecessary for the CPU 120 to take control over the data bus, its access priority is lowered then normal. As a result, this prevents the rise of the situation in which the CPU is inadvertently occupying the data bus MEM_BUS when actually it is not necessary. That is, simply

because of the rotation of the priority list happens to be pointing to the CPU 120 does not imply that the data bus would be occupied useless while some other functional module is seeking to gain control. In the depicted flow chart of FIG. 3, even though the main cycling loop of firmware routine is pointing to the CPU 120 when it is not necessary to have the data bus MEM_BUS access right granted, there will be no obstacle for other functional modules in the MPEG processor 100 to gain access over the memory resource 400. This is because that the CPU priority had been maintained at its relatively lower level. Thus, the overall MPEG decompression performance can be substantially improved over that of the prior art method.

While the invention has been described by way of example and in terms of preferred embodiment, it is to be understood that the invention need not be limited to the disclosed embodiments. On the contrary, it is intended to cover various modifications and similar arrangements included within the spirit of the appended claims, the scope of which should be accorded the broadest interpretation so as to encompass all such modifications and similar structures.

4 . B r i e f E x p l a n a t i o n o f t h e D r a w i n g s

Other objects, features and advantages of the invention will become apparent by way of the presentation of the following detailed description of the preferred but non-limiting embodiments. The description is made with reference to the accompanying drawings in which:

FIG. 1 is a block diagram showing the internal configuration of an MPEG processor;

FIG. 2 is a flow chart showing the firmware routine of a conventional MPEG processor used for controlling the implementation of the decompression operation in a fixed priority scheme; and

FIG. 3 is a flow chart showing the firmware routine in accordance with the preferred embodiment of the present invention for an MPEG processor used for controlling the implementation of the decompression operation in an adaptive-select priority scheme.

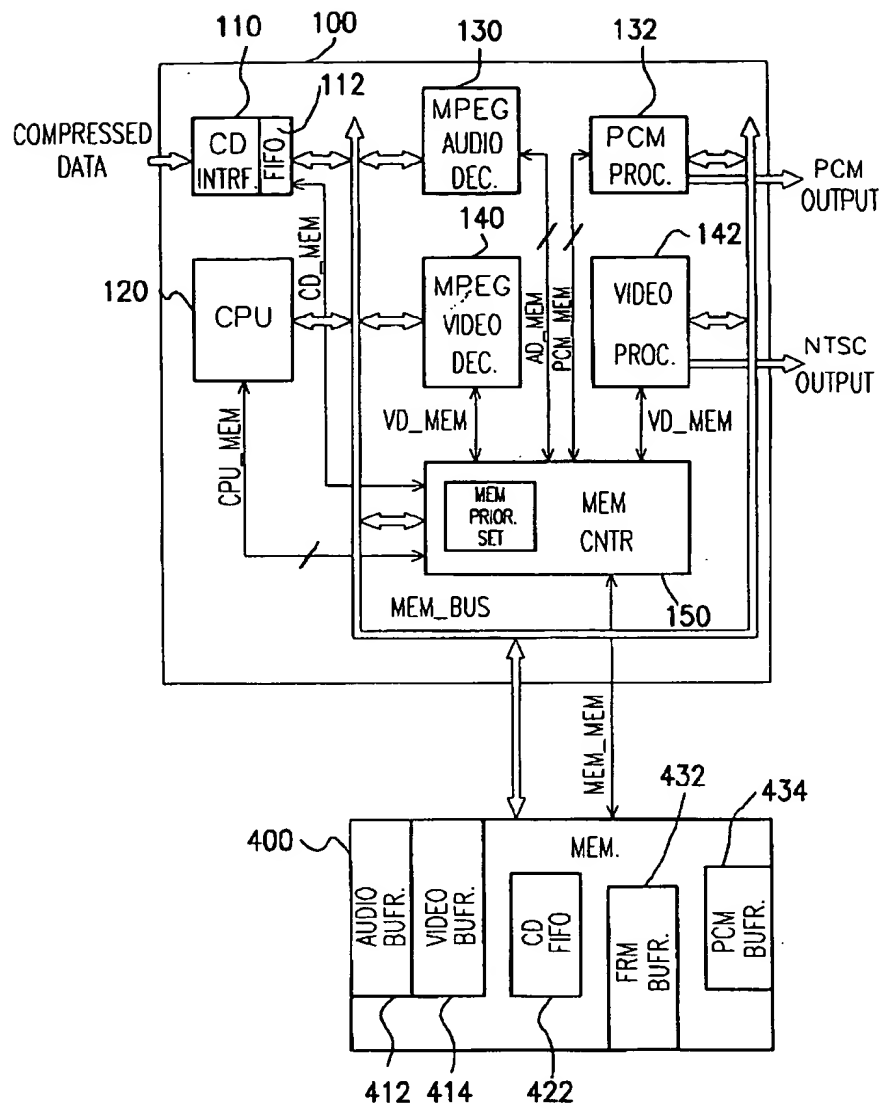


FIG. 1

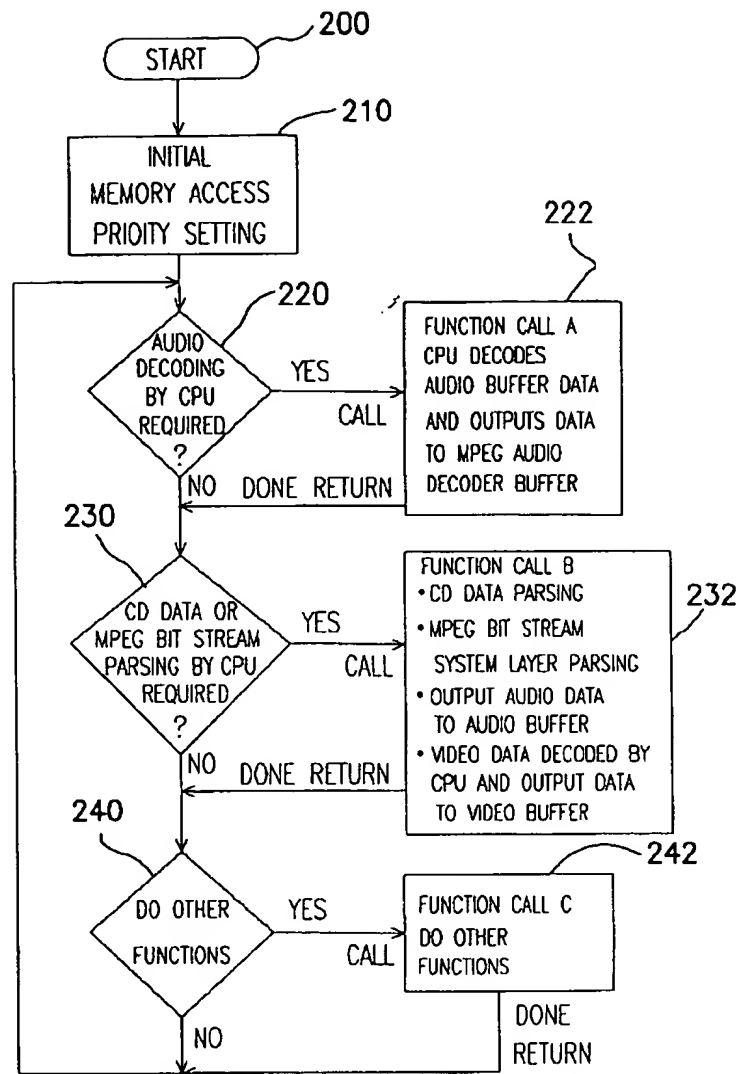


FIG. 2 (PRIOR ART)

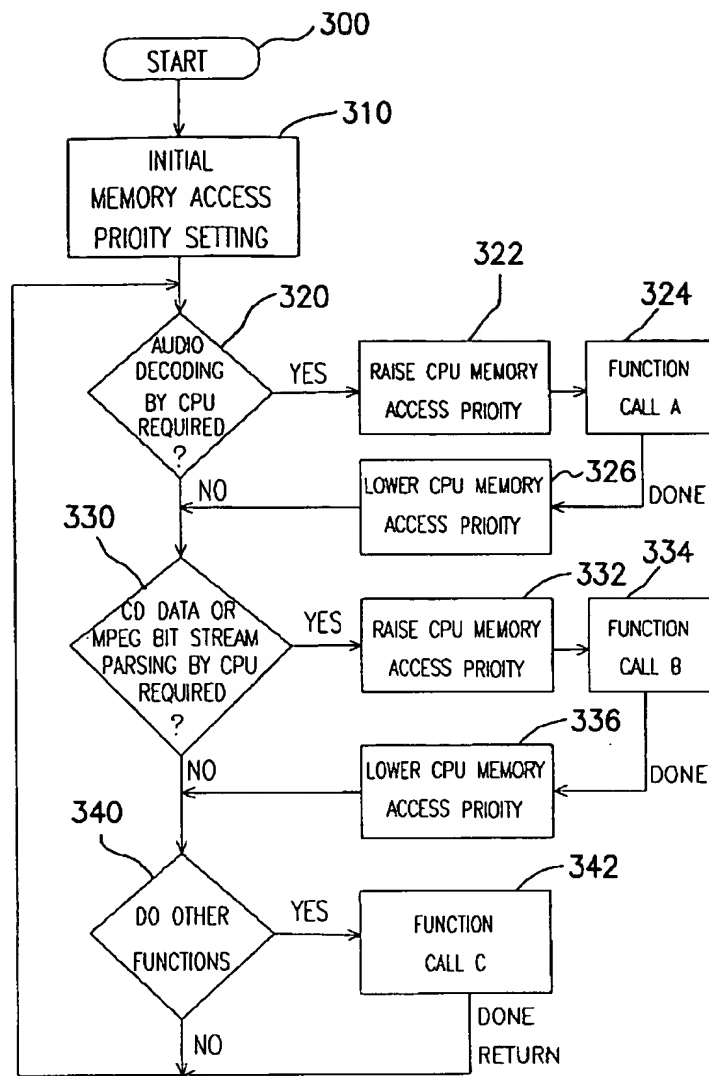


FIG. 3

1 . A b s t r a c t

An adaptive-selection method for memory access priority control in MPEG processor. The processor has functional modules that include an input interface, a CPU, an audio decoder, a video decoder, an audio processor, a video processor and a memory controller. Each of the modules gains control over the data bus via arbitration by the memory controller for accessing the memory. The access priority of the CPU to the data bus is maintained at a relatively lower level except when the CPU needs to perform parsing on the MPEG compressed data and implementing the initial decoding of the audio compressed data. The use of data bus bandwidth is therefore balanced among all the system resources thereby increasing the overall system performance.

2 . R e p r e s e n t a t i v e D r a w i n g

F i g . 1